Construction of Artificial Institutions

Maiquel de Brito

[1]UFSC-Blumenau-Brazil

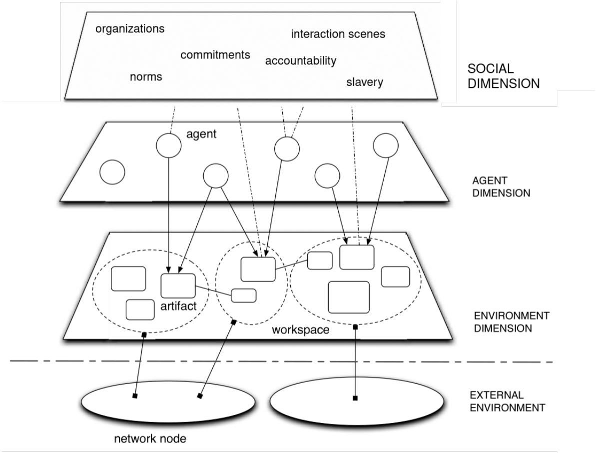
PosAutomação

June 2023

MAS: Societies of agents

It is necessary to design the society

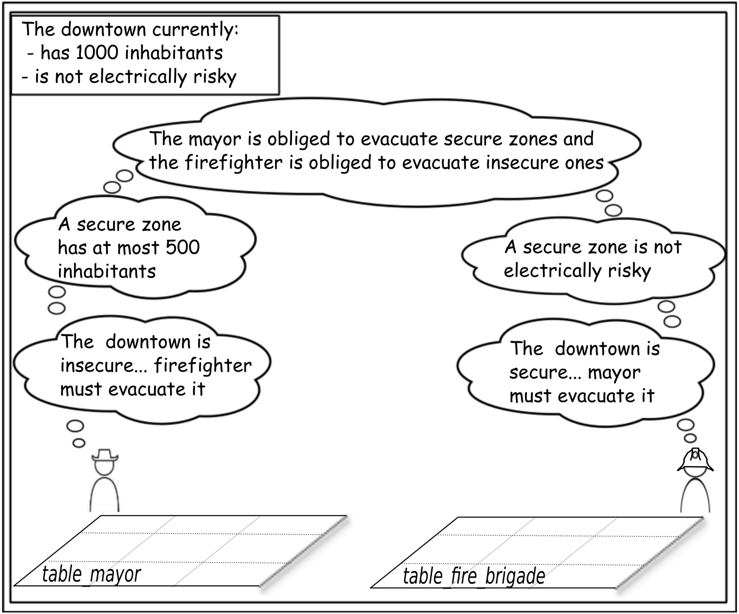
It is not enough to design the agents



Adapted from [Boissier et al. 2013]

- Organizations
  [Hübner et al. 2007]
- Norms
  [Hübner, Boissier and Bordini 2011]
- Commitments
  [Singh 1999, Dastani et al. 2015]
- Accountability
  [Baldoni et al. 2025]
- Interaction Scenes
  [García-Camino et al. 2009]
- Slavery
  [Rocha Costa 2020]

A problem [Brito et al. 2015]



Norms abstracting from environment

High level of abstraction to cover different circumstances

[Aldewereld et al. 2010]

e.g. *bidder is obliged to bid*

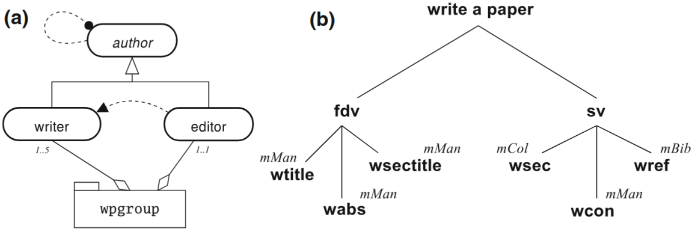
winner is obliged to pay the offer

- what is "bidder"? what is "winner"? who is obliged?
- what does it mean "to bid" and "to pay"?

Problems:

- Norm monitoring requires to connect abstract concepts to concrete elements
- Agents reasoning on norm compliance requires agents to interpret abstract concepts

Organizations abstracting from environment



(c)

mission	cardinality
mMan	1..1
mCol	1..5
mBib	1..1

(d)

role	deontic relation	mission
editor	*permission*	*mMan*
writer	*obligation*	*mCol*
writer	*obligation*	*mBib*

[Hübner et al. 2009]

- what is "writer", "editor"?
- what does it mean "to write title" and "to write abstract"?

Some questions

Questions:

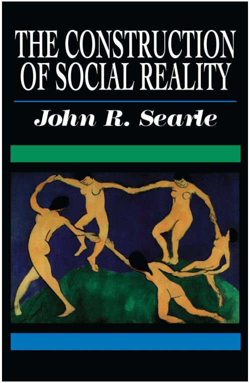
- How to represent the elements referred
  by the social abstractions (norms, organizations, etc.)?
  make them explicit

- How to connect them with the concrete environment?
  make them external

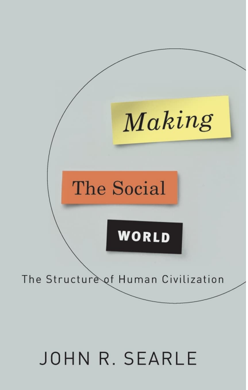
(implicit/internal notions are discussed in [Dignum et al. 2011])

Direction: the social reality (count-as) theory [Searle 1995, Searle 2009]

The social reality theory [Searle 1995, Searle 2009]



[Searle 1995]



[Searle 2009]

Institutions by Searle

"... from protons to presidents ..." [Searle 2009]

Based on the following elements:

1. Status functions
2. Collective agreement
3. Deontic powers
4. Action independent of desires and of physical constraints
5. Constitutive rules

Institutions by Searle

1. Status functions
2. Collective agreement
3. Deontic powers
4. Action independent of desires and of physical constraints
5. Constitutive rules

Functions assigned to concrete elements

not performed due to natural virtues

examples:

- piece of paper counts as five dollar bill
- individual counts as president
- raising of hands counts as bid

Institutions by Searle

1. Status functions
2. Collective agreement
3. Deontic powers
4. Action independent of desires and of physical constraints
5. Constitutive rules

People agree about (or at least accept)

the functions performed

by the concrete elements (that are not inherent to them)

Institutions by Searle

1. Status functions
2. Collective agreement
3. Deontic powers
4. Action independent of desires and of physical constraints
5. Constitutive rules

Express what people are expected to do or to avoid

rights, duties, obligations, authorizations, permissions, etc.

Refer to status functions

instead of to refer to concrete elements

- a PhD student is obliged to finish his thesis on time
- a PhD student is permitted to finish his thesis before the final time
- a PhD student is prohibited to committing any plagiarism in his thesis

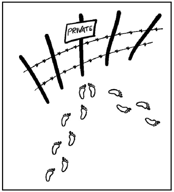
Institutions by Searle

Status Functions + Deontic Powers:

"the glue that holds the human civilization together" [Searle 2009] - page 9

1. Status functions
2. Collective agreement
3. Deontic powers
4. Action independent of desires and of physical constraints
5. Constitutive rules

provide reasons to people act independent of
- their own desires
- physical constraints implemented in the brute reality



Institutions by Searle

1. Status functions
2. Collective agreement
3. Deontic powers
4. Action independent of desires and of physical constraints
5. Constitutive rules

Define how status functions are constituted

connecting them to the brute reality

X counts as Y in C

e.g. A fence (X) counts as
a boundary of private property (Y)
if the property is registered to some owner (C)

Institutions by Searle

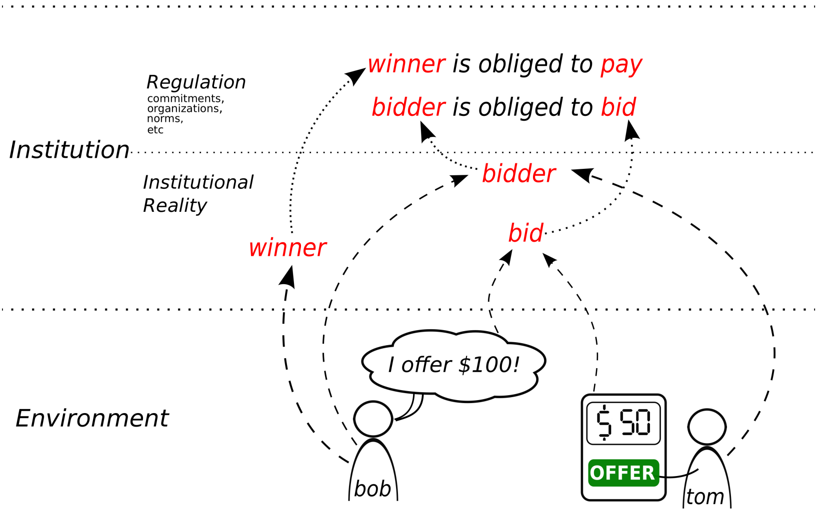
"...institutions are systems of constitutive rules..."

(SEARLE, 2009, page 10)

Constituted status functions

- the building block of institutions
- constituted thanks to constitutive rules
- deontic powers are attached to
- required to regulation: individuals behave as expected when status functions are constituted

Institutions in Agent Societies



Situated Artificial Institution (SAI)

Institution: Regulation + Institutional reality

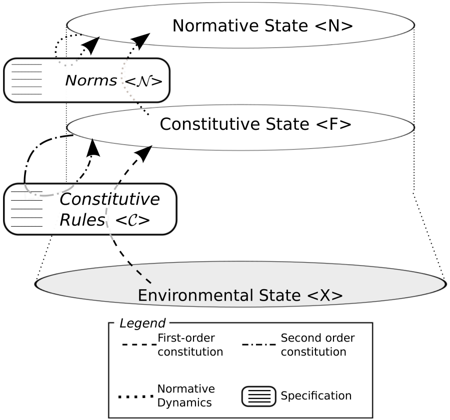
Assumption: regulation provided by existing [normative] models

existing works are enough

What is needed?

1. A model of institutional reality
2. Coupling existing norms in the institutional reality

Situated Artificial Institution (SAI) [Brito et al. 2018]



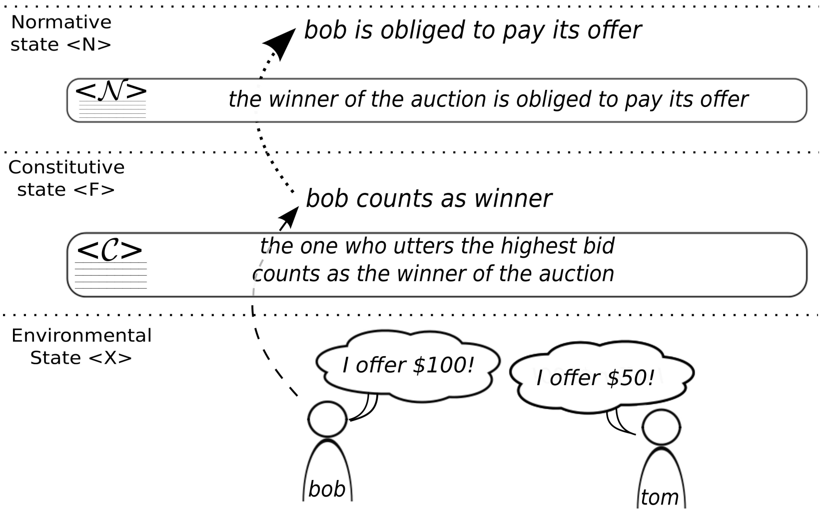
Specification

- Norms ($\mathcal{N}$):
  - define what agents must do/avoid
  - composed of status functions
- Status functions ($\mathcal{F}$): functions assignable to agents, events and states
- Constitutive rules ($\mathcal{C}$): constitute the *status functions*

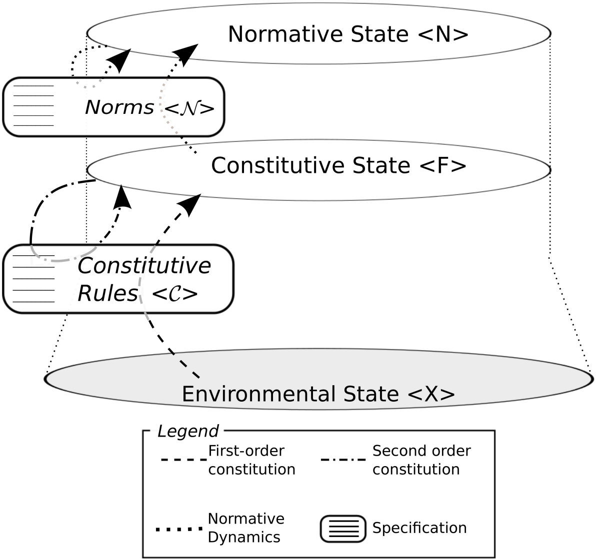
Dynamics

- Normative state (N): expected agents' behaviour
- Constitutive state (F): institutional reality

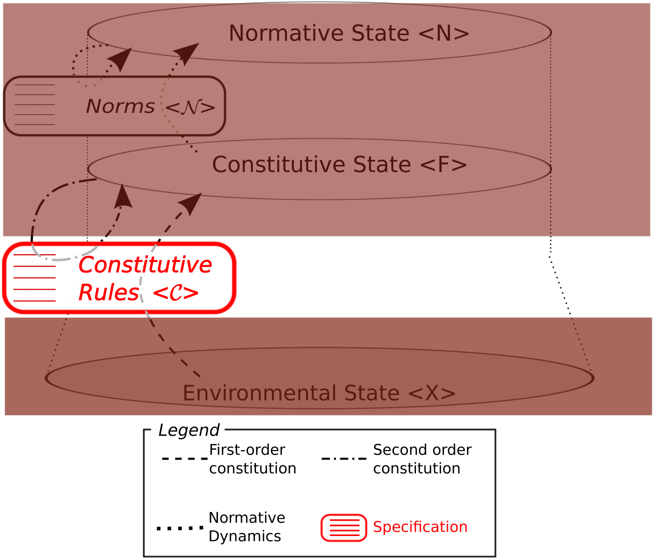
Situated Artificial Institution (SAI)



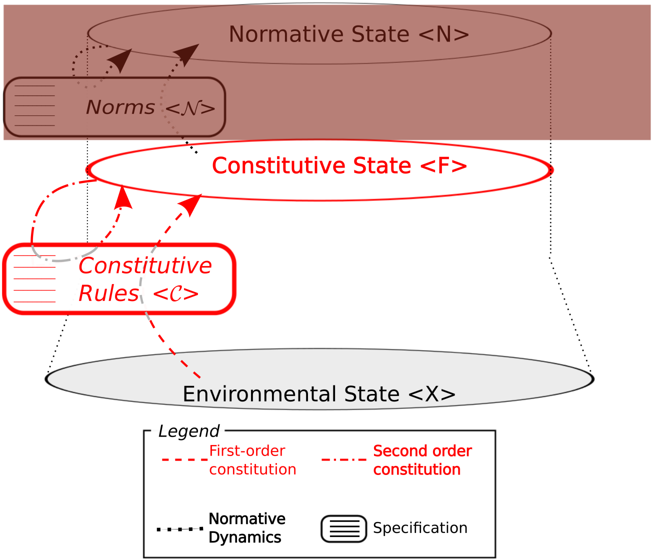
Institutional Reality in SAI



Institutional Reality in SAI



Institutional Reality in SAI



Institutional Reality in SAI - Constitutive Specification

Status functions: status assigned to agents, events, and states
that impose functions to them

Status functions in SAI: $\mathcal{F} = \mathcal{A}_\mathcal{F} \cup \mathcal{E}_\mathcal{F} \cup \mathcal{S}_\mathcal{F}$ s.t.

- $\mathcal{A}_\mathcal{F}$: status functions assignable to **agents**, i.e. agent-status functions
  e.g.: bob counts as winner

- $\mathcal{E}_\mathcal{F}$: status functions assignable to **events**, i.e. event-status functions
  e.g.: tell("I offer $100") counts as bid

- $\mathcal{S}_\mathcal{F}$: status functions
  - assignable to **states**, i.e. state-status functions
    e.g.: "twenty agents in the system" counts as quorum for an auction
  - involved in freestanding assignment
    e.g. assigned to conditions that do not involve environmental states

Institutional Reality in SAI - Constitutive Specification

Constitutive rules: specify the constitution of status functions

Constitutive rules in SAI: $c \in \mathcal{C} = \langle x, y, t, m \rangle$ s.t.

- x has the *status function* y when t occurs, while m holds
- t and m can be omitted

Example:

$$\langle Agent, auctioneer, (propose(auction), Agent), \neg auction\_finished \rangle$$

$\downarrow$ expressed as $\downarrow$

```
Agent count-as auctioneer
      when (propose(auction),Agent)
      while not auction_finished.
```

Institutional Reality in SAI - Constitutive Specification

```
status_functions:
   agents: firefighter
   events: evacuate(Zone)
   states: secure(Zone)

constitutive_rules:
   /* Actors carry the status functions    according to their check in the tables */
   Agent count-as firefighter
       when checkin(table_fire_brigade,Agent).

   /* A zone preventive phase of crisis managament counts as    that zone being secure if
   (i) it does not pose electrical risks and (ii) it has at most 500 inhabitants */
   security_phase(_,Zone,preventive) count-as secure(Zone)
      while not( AnyState is electric_risky(Zone)) &
            ((nbInhabit(_,Zone,X)& X<=500) |
            security_phase(_,Zone,preventive) is secure(Zone)).

   /* Firefighter putting the object launch_tangible in the
      some coordinates of a table counts as the evacuation of a zone */
   putTangible(_,launch_tangible,1,2)[sai__agent(Actor)]
      count-as evacuate(downtown)
      while Actor is firefighter.

   putTangible(_,launch_tangible,3,3)[sai__agent(Actor)]
      count-as evacuate(industrial_zone)
      while Actor is firefighter.
```

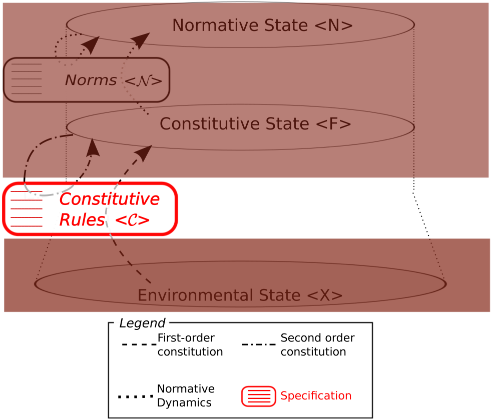
Institutional Reality in SAI - Constitutive dynamics

Interpretation of constitutive rules

- Produces constitution of status functions

Constitutive state (F)

- Set of constituted status functions
- the SAI representation of institutional reality



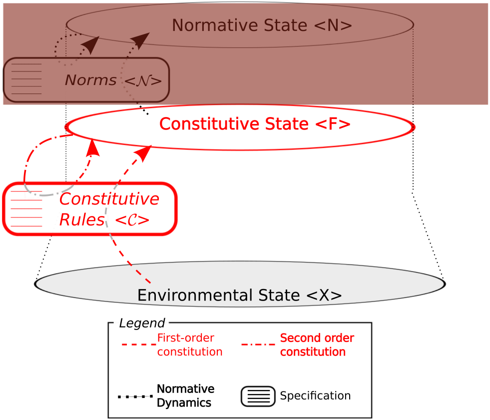
Institutional Reality in SAI - Constitutive dynamics

Interpretation of constitutive rules

- Produces constitution of status functions

Constitutive state (F)

- Set of constituted status functions
- the SAI representation of institutional reality



Constitutive dynamics - 1^{st} and 2^{nd} order constitution

- First-order constitution

  Status functions assigned to environmental elements

- Second-order constitution

  Status functions assignments lead to new status functions assignments

Details on constitutive dynamics are in [Brito et al. 2015b, Brito et al. 2018]

Constitutive dynamics - 1^{st} and 2^{nd} order constitution

- **Environmental state :** $X = A_X \cup E_X \cup S_X$

  where A_X =agents acting; E_X=events occurring; S_X = states holding

- **Constitutive state :** $F = A_F \cup E_F \cup S_F$

  where A_F =agent-SFAs ; E_X=event-SFAs; S_X = state-SFAs

1^{st} order constitution of agent-status functions

$$f\text{-}const_a(\mathcal{F}, \mathcal{C}, X^0 \cdots X^z, F^0 \cdots F^z, i) = \{\langle x, y\rangle | x \in A^i_X \wedge y \in \mathcal{A_F} \wedge \exists c \in \mathcal{C}\ \exists \theta\ \exists s \in \mathbb{N}\ \forall k \in [s, i] : (E^s_X \cup\ E^s_F \models t\theta) \wedge (X^k \cup F^k \models m\theta) \wedge x'\theta = x\}$$

$$s.t.\ c = \langle x', y, t, m\rangle$$

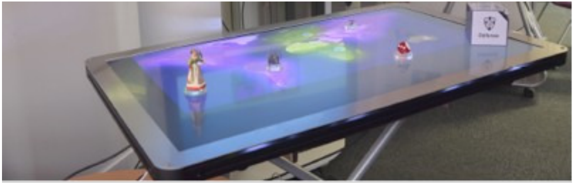
2^{nd} order constitution of agent-status functions

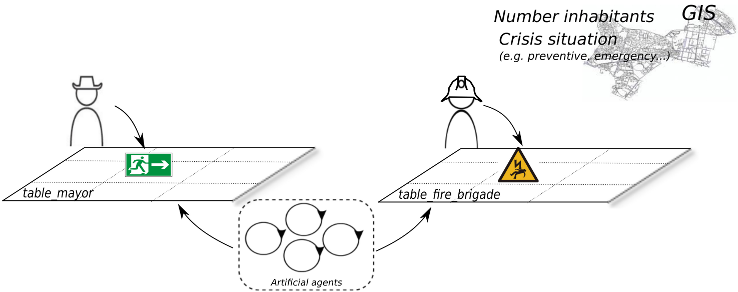
$$s\text{-}const_a(\mathcal{F}, \mathcal{C}, X^0 \cdots X^z, F^0 \cdots F^z, i) = \{\langle x, y\rangle | x \in A^i_X \wedge y \in \mathcal{A_F} \wedge \exists c \in \mathcal{C}\ \exists \theta\ \exists s \in \mathbb{N}\ \forall k \in [s, i] : (E^s_X \cup\ E^s_F \models t\theta) \wedge (X^k \cup F^k \models m\theta) \wedge x'\theta \in \mathcal{A_F} \wedge \langle x, x'\theta\rangle \in A^i_F\}$$

$$s.t.\ c = \langle x', y, t, m\rangle$$

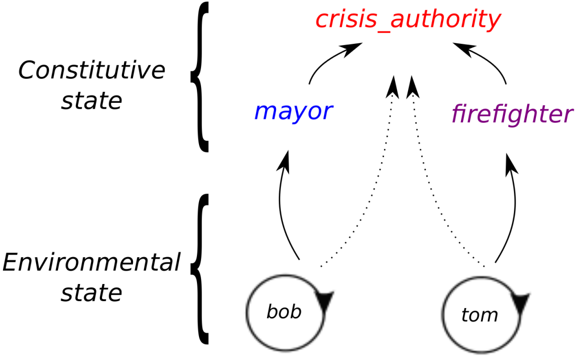
Constitutive dynamics - example scenario

crisis management





Constitutive dynamics - 1^{st} and 2^{nd} order constitution

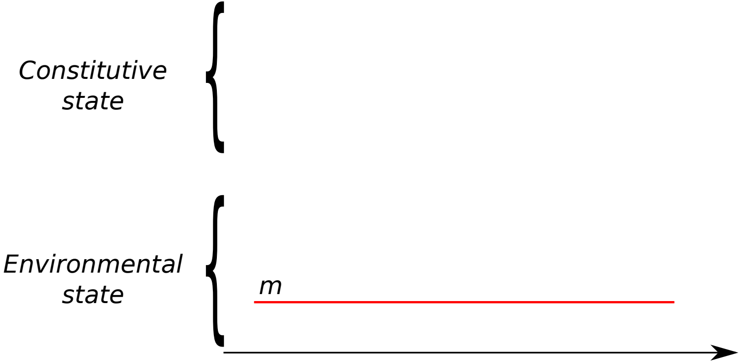


1^{st} order { *bob count as* *mayor*
tom count as *firefighter*

2^{nd} order { *mayor count-as* *crisis_authority*
firefighter count-as *crisis_authority*

Constitutive dynamics - 1^{st} order constitution

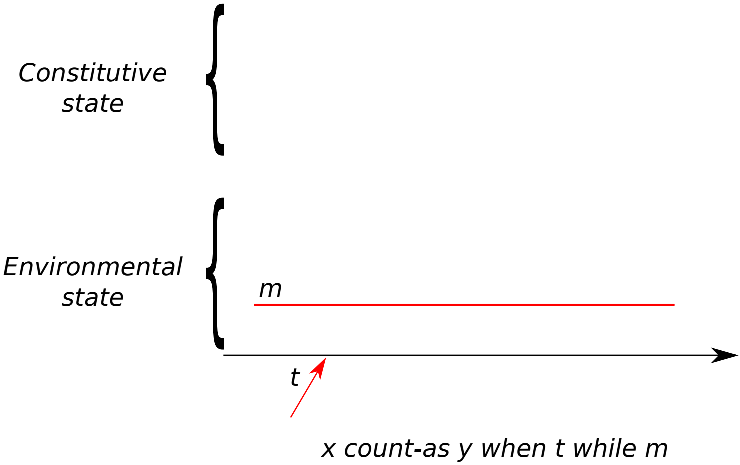
Agent- and state-status function



x count-as y when t while m

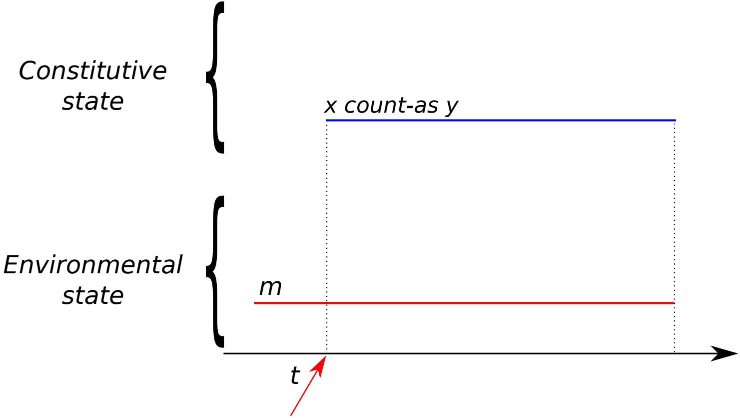
Constitutive dynamics - 1^{st} order constitution

Agent- and state-status function



Constitutive dynamics - 1^{st} order constitution

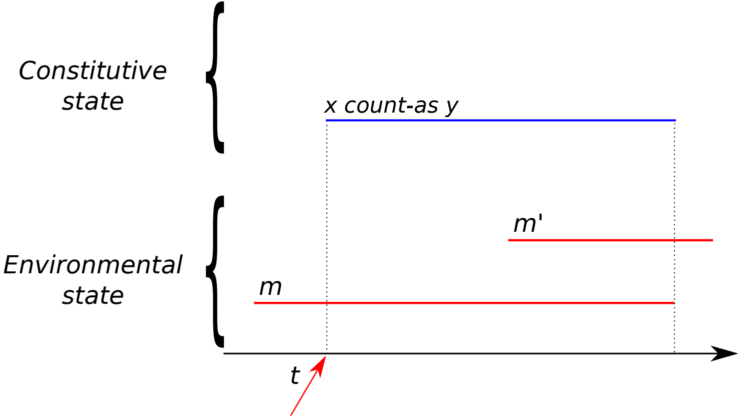
Agent- and state-status function



x count-as y when t while m

Constitutive dynamics - 2^{nd} order constitution

Agent- and state-status function

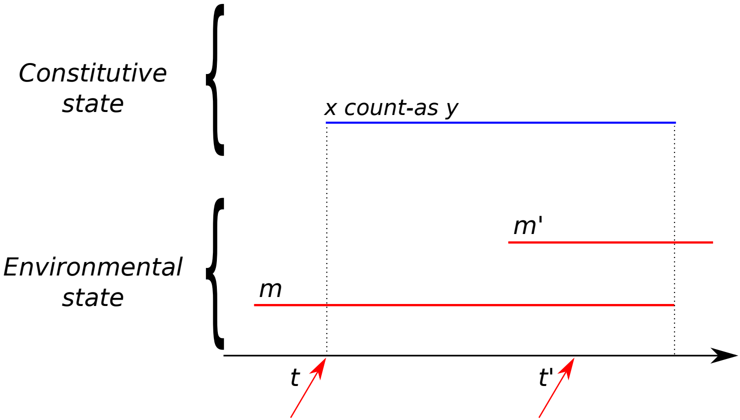


x count-as y when t while m

y count-as y' when t' while m'

Constitutive dynamics - 2^{nd} order constitution

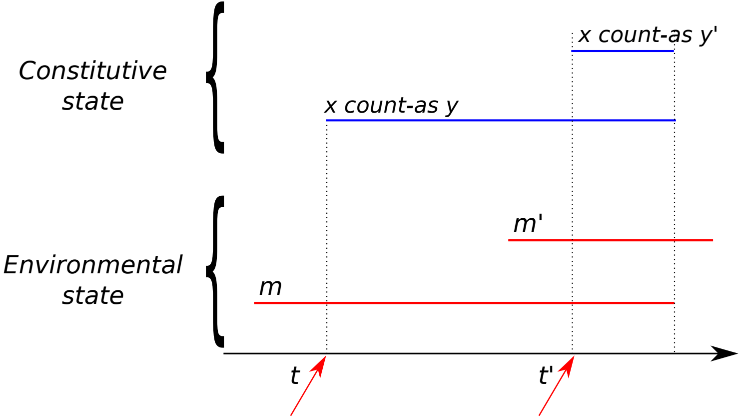
Agent- and state-status function



x count-as y when t while m
y count-as y' when t' while m'

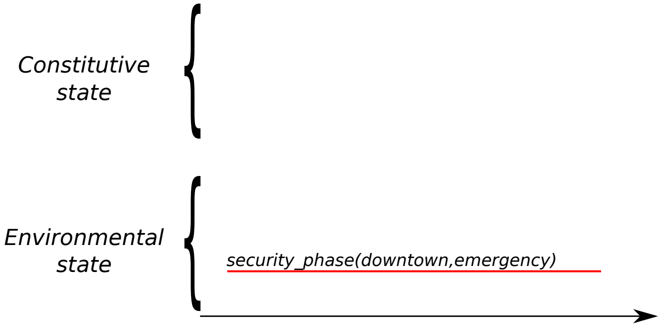
Constitutive dynamics - 2^{nd} order constitution

Agent- and state-status function



x count-as y when t while m
y count-as y' when t' while m'

Constitutive dynamics - Example



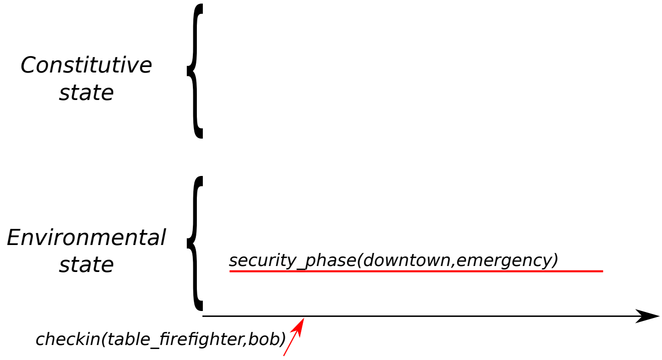
1st order

```
Agent count-as firefighter
  when checkin(table_firefighter,Agent)
  while security_phase(downtown,emergency).
```

2nd order

```
firefighter
  count-as crisis_authority
  when command_evacuation(_).
```

Constitutive dynamics - Example



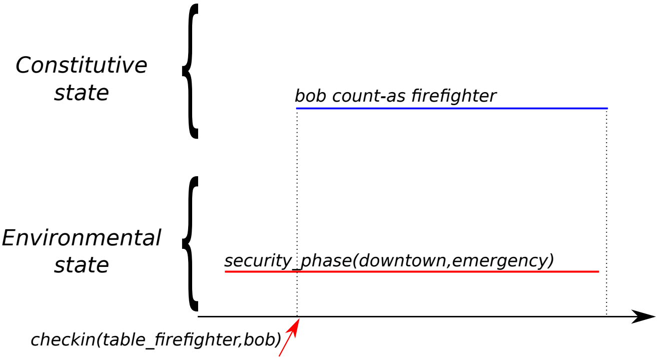
1st order

```
Agent count-as firefighter
  when checkin(table_firefighter,Agent)
  while security_phase(downtown,emergency).
```

2nd order

```
firefighter
  count-as crisis_authority
  when command_evacuation(_).
```

Constitutive dynamics - Example



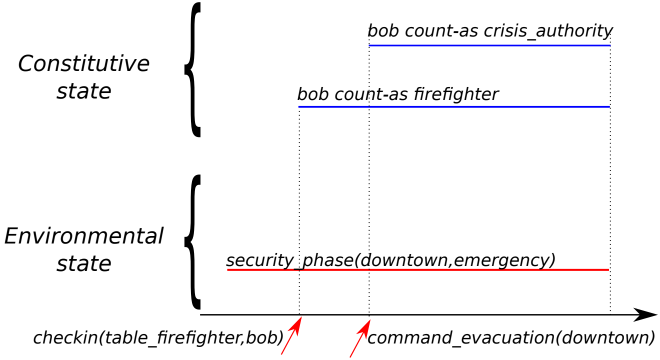
1st order

`Agent` **count-as** `firefighter`
  when `checkin(table_firefighter,Agent)`
  while `security_phase(downtown,emergency).`

2nd order

`firefighter`
  count-as `crisis_authority`
  when `command_evacuation(_).`

Constitutive dynamics - Example



1st order

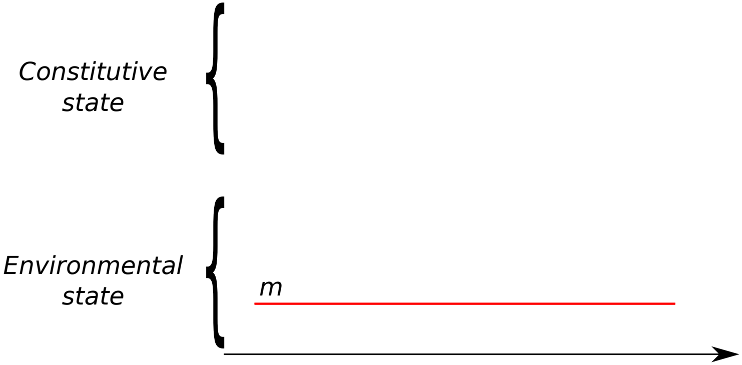
Agent **count-as** firefighter
  when checkin(table_firefighter,Agent)
  while security_phase(downtown,emergency).

2nd order

firefighter
  count-as crisis_authority
  when command_evacuation(_).

Constitutive dynamics - 1^{st} order constitution

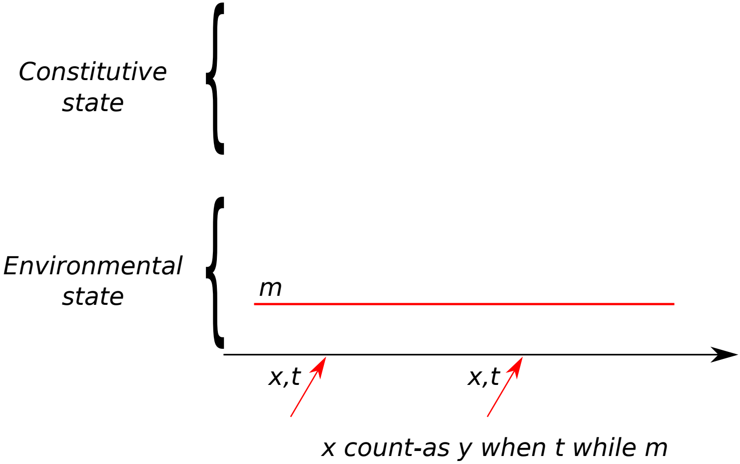
Event-status function



x count-as y when t while m

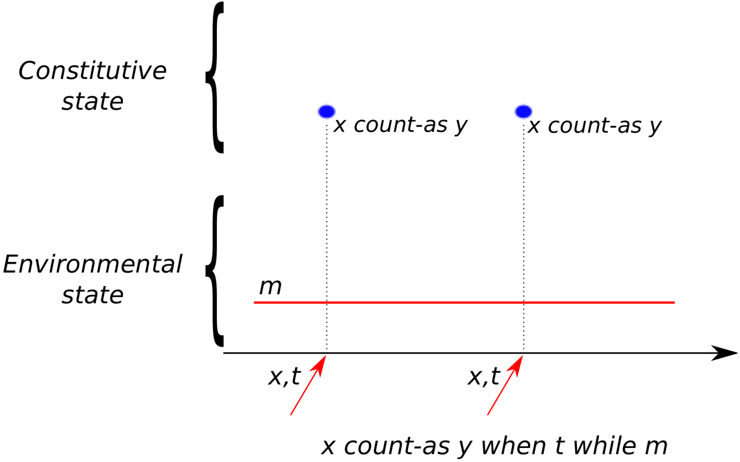
Constitutive dynamics - 1^{st} order constitution

Event-status function



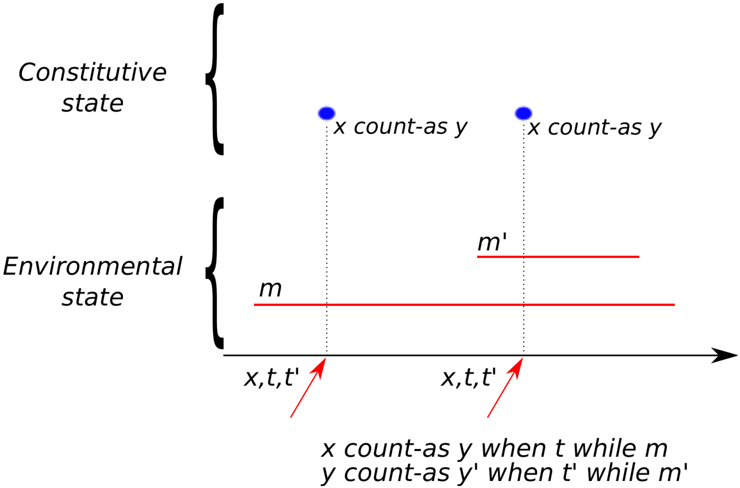
Constitutive dynamics - 1^{st} order constitution

Event-status function



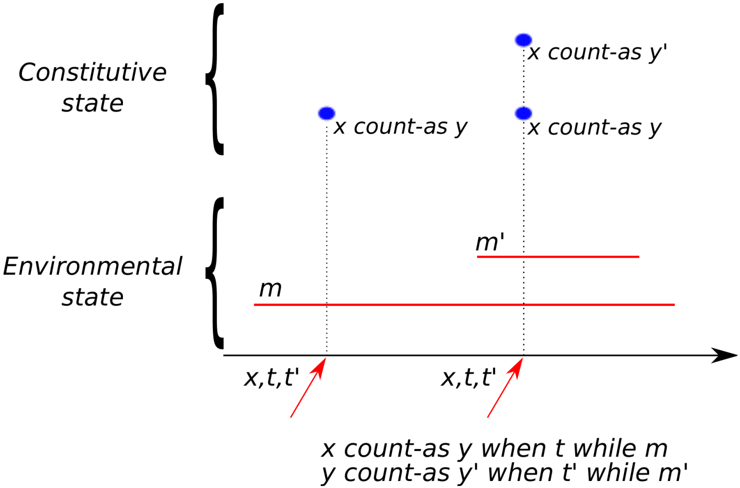
Constitutive dynamics - 2^{nd} order constitution

Event-status function

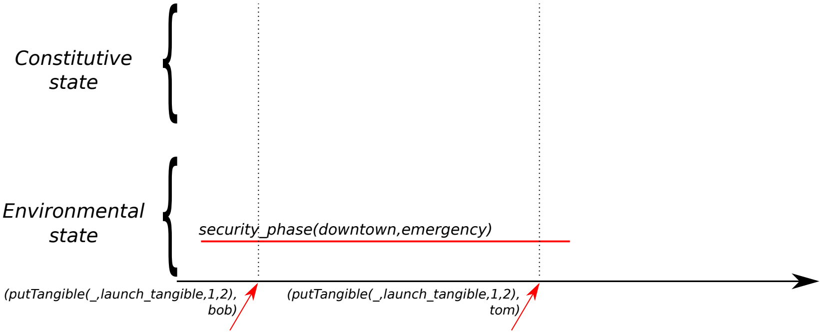


Constitutive dynamics - 2^{nd} order constitution

Event-status function



Constitutive dynamics - example



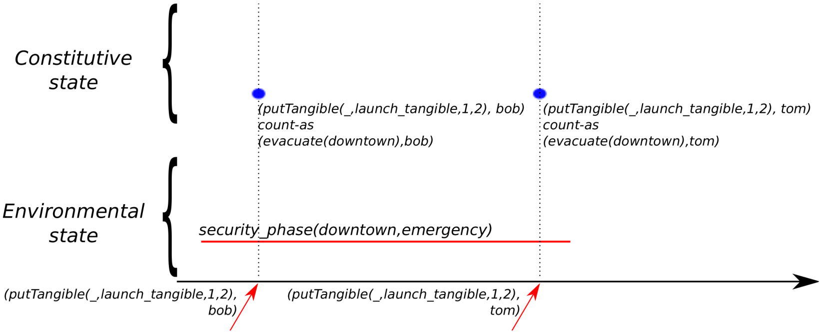
1st order

putTangible(_,launch_tangible,1,2)
[sai__agent(_)]
count-as evacuate(downtown)
while security_phase(downtown,emergency).

2nd order

evacuate(downtown)
count-as security_procedure.

Constitutive dynamics - example



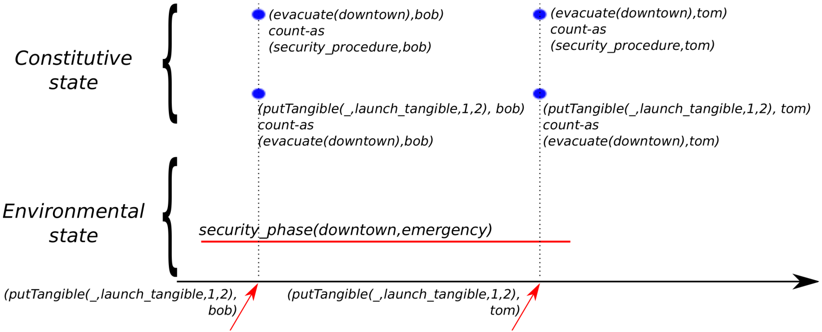
1st order

```
putTangible(_,launch_tangible,1,2)
                    [sai__agent(_)]
  count-as evacuate(downtown)
  while security_phase(downtown,emergency).
```

2nd order

```
evacuate(downtown)
  count-as security_procedure.
```

Constitutive dynamics - example



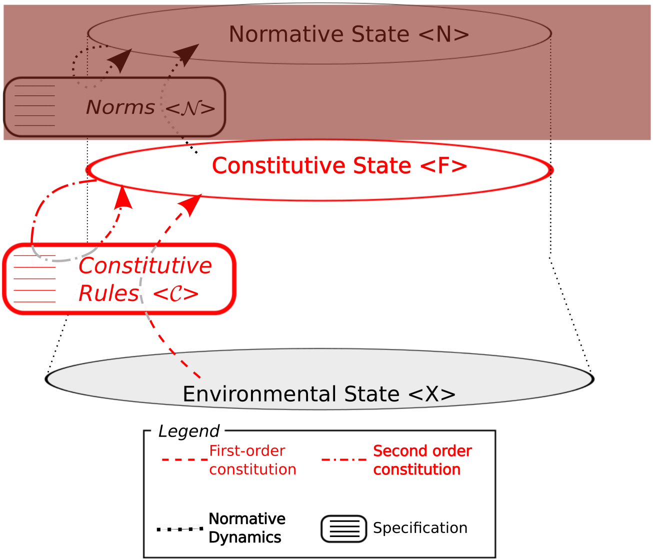
1st order

```
putTangible(_,launch_tangible,1,2)
                   [sai__agent(_)]
  count-as evacuate(downtown)
  while security_phase(downtown,emergency).
```

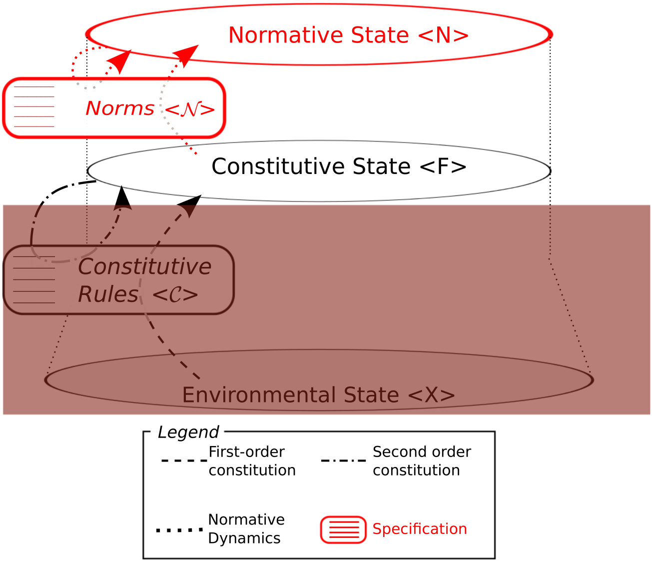
2nd order

```
evacuate(downtown)
  count-as security_procedure.
```

Regulation in SAI



Regulation in SAI



Regulation in SAI

[Brito et al. 2019, Brito et al. 2018]

Coupling norms with the constitutive state

(considering their specific representations and dynamics)

Requirements:

1. Normative representation:

   to introduce the status functions in the norm representation

2. Normative dynamics

   to define how the different components of the norms are interpreted

Regulation in SAI

[Brito et al. 2019, Brito et al. 2018]

Coupling norms with the constitutive state

(considering their specific representations and dynamics)

Requirements:

1. **Normative representation**:
   to introduce the status functions in the norm representation
2. Normative dynamics
   to define how the different components of the norms are interpreted

Regulation in SAI

[Brito et al. 2019, Brito et al. 2018]

Coupling norms with the constitutive state

(considering their specific representations and dynamics)

Requirements:

1. Normative representation:
   to introduce the status functions in the norm representation

2. **Normative dynamics**
   to define how the different components of the norms are interpreted

Norm representation in SAI

Introducing status functions in the norms

A generic approach:

based on the ADICO model [Crawford e Ostrom 1995]

- **A** $\in \mathcal{A}_{\mathcal{F}}$
- **I** $\in \mathcal{E}_{\mathcal{F}} \cup \mathcal{S}_{\mathcal{F}}$
- **C** $\in \mathcal{W}_{\mathcal{F}}$

where

- **A** is the agent targetted by the norm
- **I** is the prescribed outcome
- **C** are the conditions under which the norm must be followed

Normative dynamics with SAI

Different normative models have particular dynamics

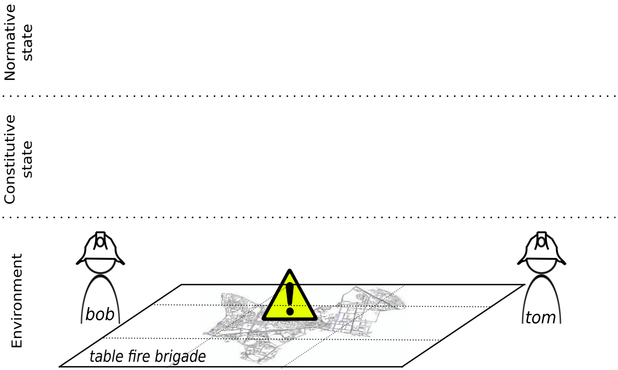
Difficult to generalise but...
... all the norms can be at least activated and fulfilled

In the sequence: some details on norm activation and fulfilment
considering norms by [Panagiotidi et al., 2013]

Activation

– Norm instances are created when activation condition is satisfied. An instance for every agent carrying α

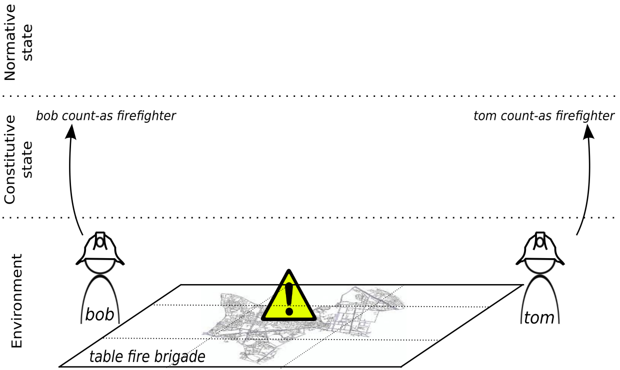
```
norm: insecure(Zone) -> obligation(firefighter,norm,evacuate(Zone),'now'+'1 day').
```



Activation

- Norm instances are created when activation condition is satisfied. An instance for every agent carrying α

`norm: insecure(Zone) -> obligation(firefighter,norm,evacuate(Zone),'now'+'1 day').`



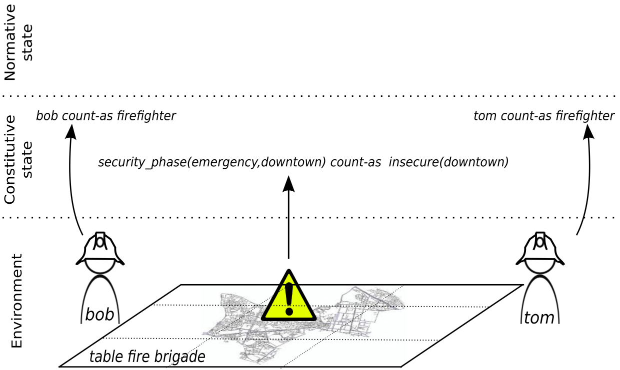
Agent **count-as** **firefighter**
when checkin(table_fire_brigade,Agent).

Environmental events:
checkin(table_fire_brigade,bob)
checkin(table_fire_brigade,tom)

Activation

- Norm instances are created when activation condition is satisfied. An instance for every agent carrying α

`norm: insecure(Zone) -> obligation(firefighter,norm,evacuate(Zone),'now'+'1 day').`



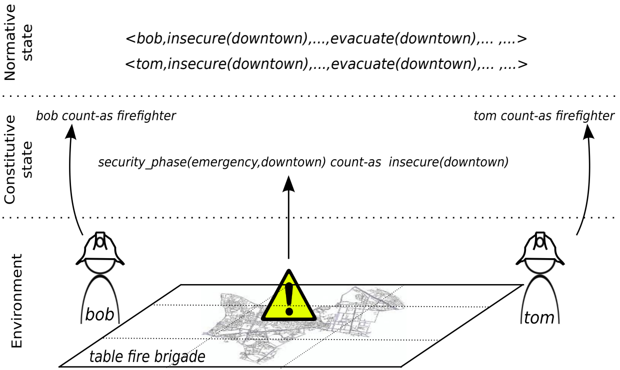
`security_phase(emergency,Zone)`
count-as insecure(Zone).

Environmental state:
`security_phase(emergency,downtown)`

Activation

– Norm instances are created when activation condition is satisfied. An instance for every agent carrying α

```
norm: insecure(Zone) -> obligation(firefighter,norm,evacuate(Zone),'now'+'1 day').
```



Activation

Formally:

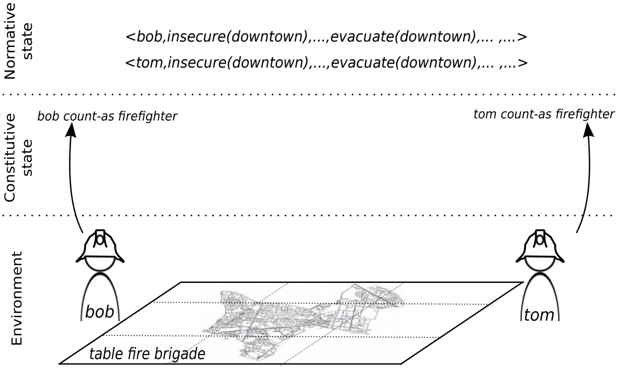
$$activated(\mathcal{N}, F, N) = \{n' | \exists\theta\exists\langle\alpha, c_a, c_m, c_d, c_r, c_t\rangle \in \mathcal{N} : \\ F \cup N \models c_a\theta \wedge (a_X \textbf{ is } \alpha\theta) \wedge n' \notin AS\}$$

$$\text{s.t.} \quad n' = \langle(a_X, \alpha\theta), c_a\theta, c_m\theta, c_d\theta, c_r\theta, c_t\theta\rangle$$

Fulfilments

Individual agent level approach for events

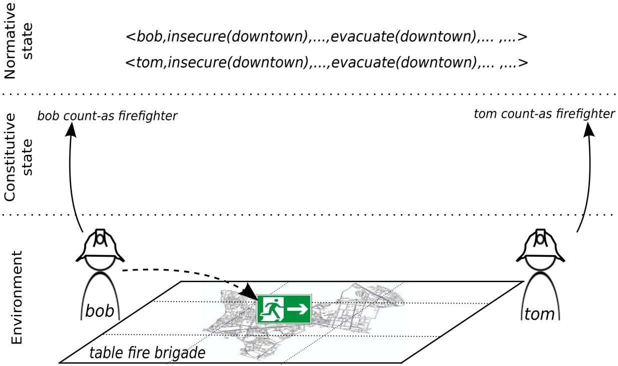
(`evacuate(.)` is an event status function)



Fulfilments

Individual agent level approach for events

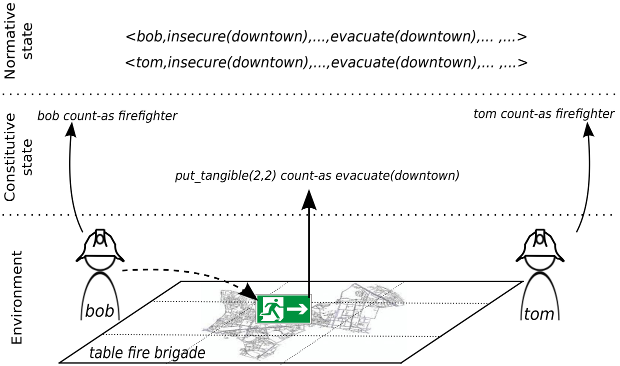
(`evacuate(.)` is an event status function)



Fulfilments

Individual agent level approach for events

(`evacuate(.)` is an event status function)



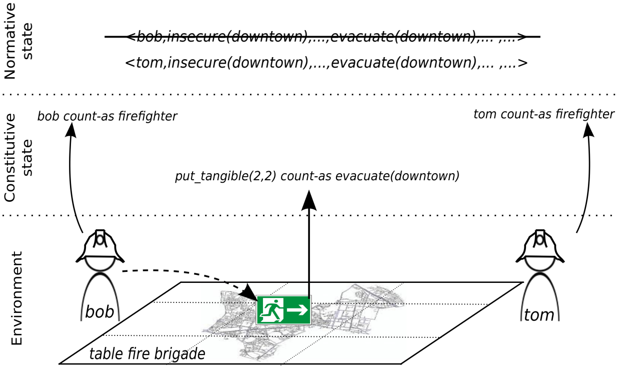
Environmental events:
`put_tangible(2,2,bob)`

`put_tangible(2,2,Agent)`
count-as **evacuate(downtown)**.

Fulfilments

Individual agent level approach for events

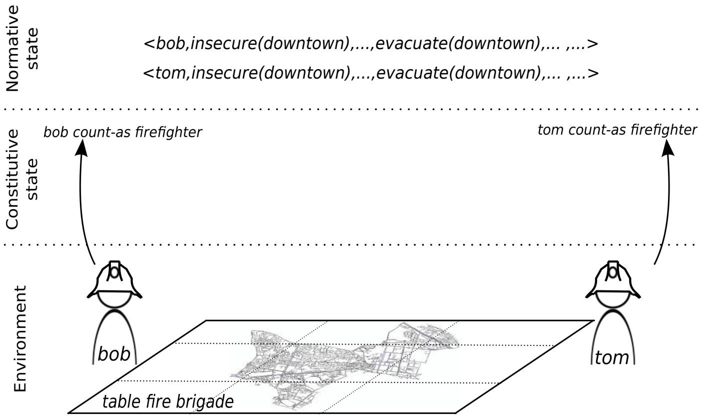
(`evacuate(.)` is an event status function)



Fulfilments

Individual agent level approach for events

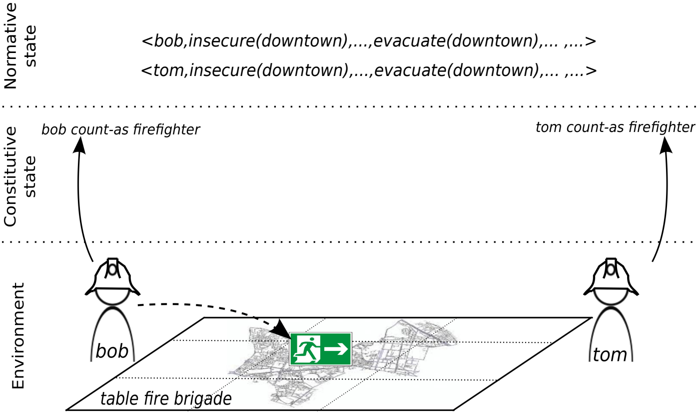
(`evacuate(.)` is an event status function)



Fulfilments

Individual agent level approach for events

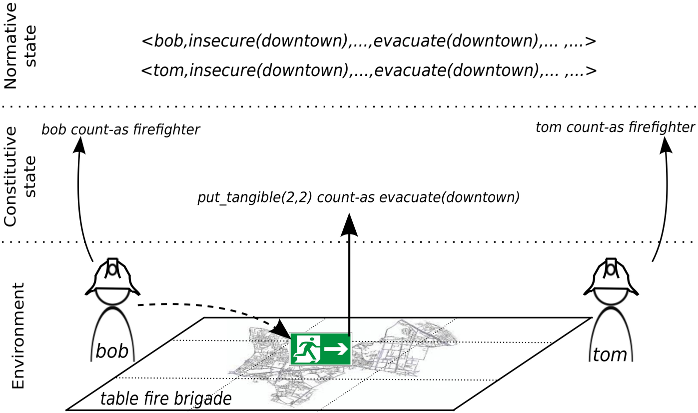
(`evacuate(.)` is an event status function)



Fulfilments

Individual agent level approach for events

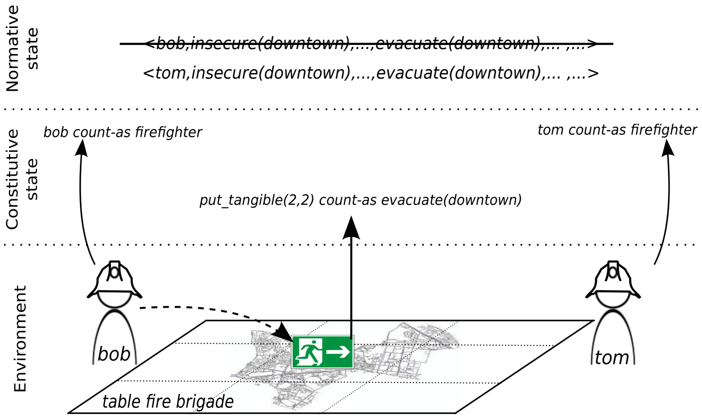
(`evacuate(.)` is an event status function)



Fulfilments

Individual agent level approach for events

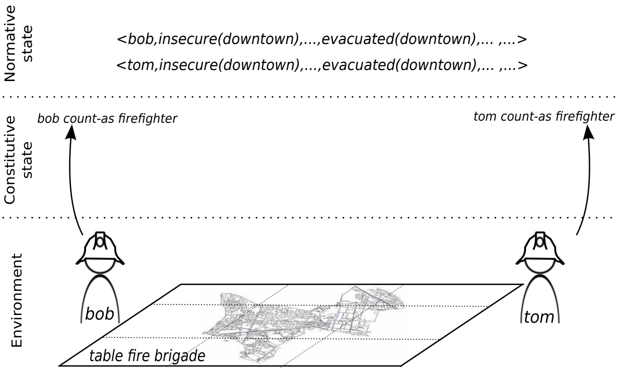
(`evacuate(.)` is an event status function)



Fulfilments

See to it approach for states

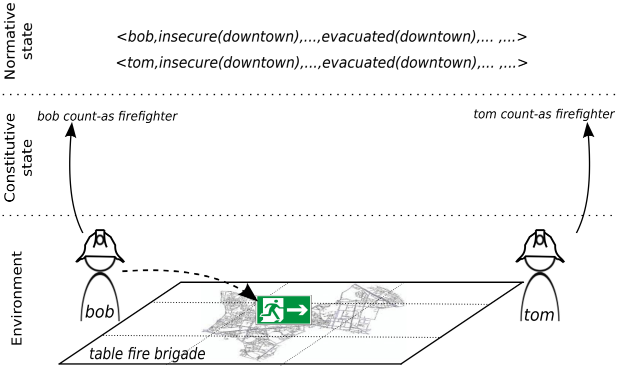
(`evacuated(.)` is a state status function)



Fulfilments

See to it approach for states

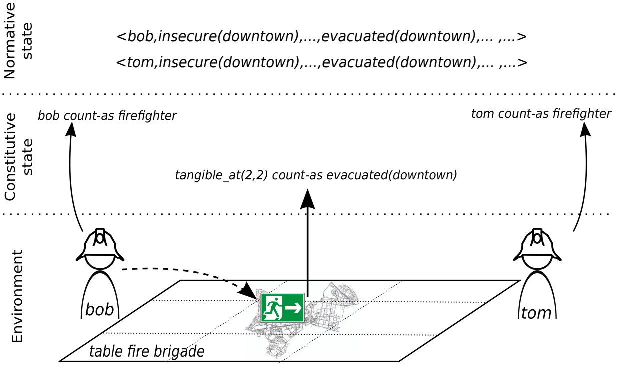
(`evacuated(.)` is a state status function)



Fulfilments

See to it approach for states

(`evacuated(.)` is a state status function)



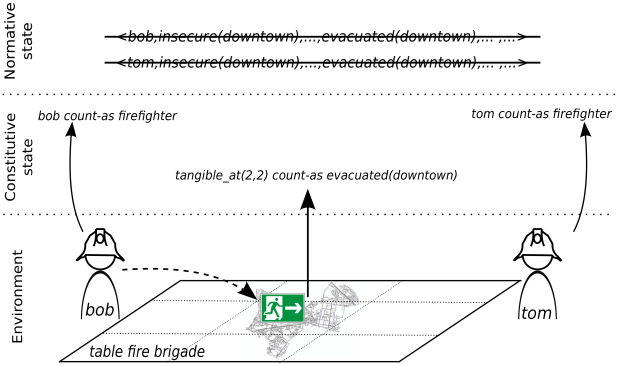
`tangible_at(2,2)`
count-as **evacuated(downtown)**.

Environmental state:
`tangible_at(2,2)`

Fulfilments

See to it approach for states

(`evacuated(.)` is a state status function)



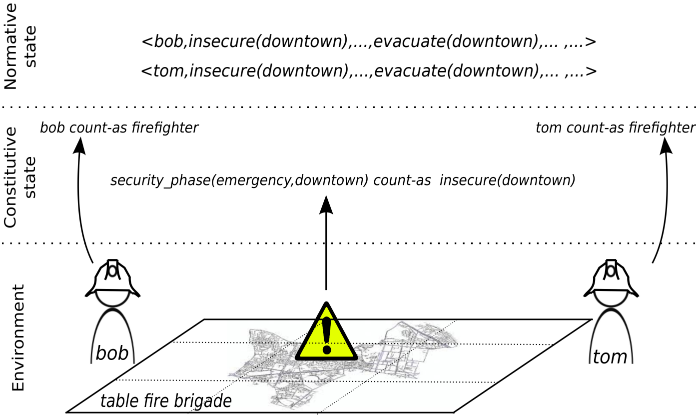
`tangible_at(2,2)`
count-as **evacuated(downtown)**.

Environmental state:
`tangible_at(2,2)`

Fulfilments

Revocation of agent-status function assignment

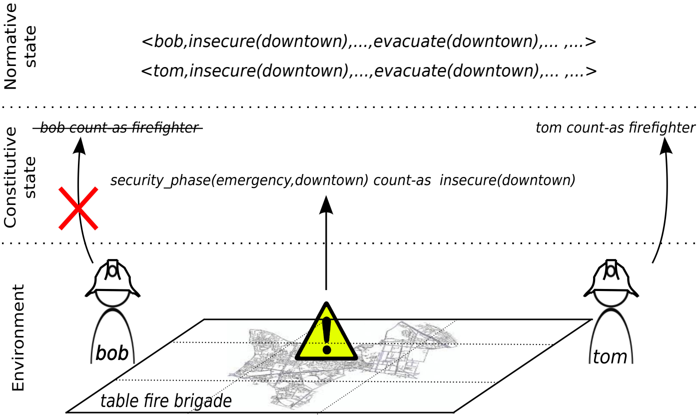
e.g. $\langle \mathit{firefighter}, \mathit{insecure}(\mathit{Zone}), \cdots, \mathit{evacuate}(\mathit{Zone}), \cdots, \cdots \rangle$



Fulfilments

Revocation of agent-status function assignment

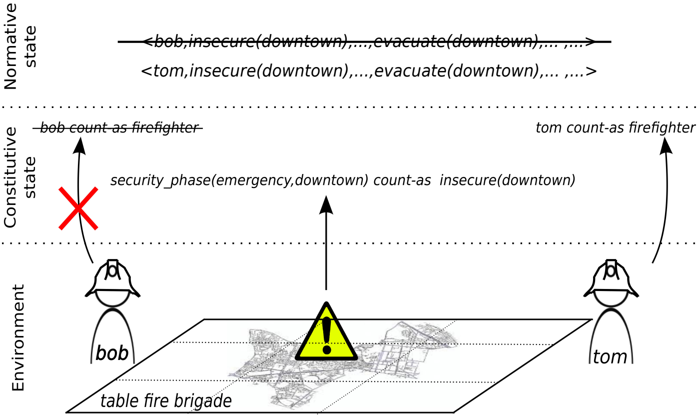
e.g. $\langle \mathit{firefighter}, \mathit{insecure}(\mathit{Zone}), \cdots, \mathit{evacuate}(\mathit{Zone}), \cdots, \cdots \rangle$



Fulfilments

Revocation of agent-status function assignment

e.g. $\langle \mathit{firefighter}, \mathit{insecure}(\mathit{Zone}), \cdots, \mathit{evacuate}(\mathit{Zone}), \cdots, \cdots \rangle$



Fulfilments

Formally:

$$f\text{-}deactivated^e(F,N) = \quad \{\langle n' | \exists (e_X, a_X) \in E_X : n' \in AS \wedge c'_d \in \mathcal{E}_\mathcal{F} \wedge$$
$$F \models ((e_X, a_X) \text{ is } c'_d \vee \neg(a_X \text{is } \alpha)) \wedge F \cup N \models c'_m\}$$

$$f\text{-}deactivated^s(F,N) = \quad \{\langle n' | n' \in AS \wedge c'_d \in \mathcal{S}_\mathcal{F} \wedge$$
$$F \models (c'_d \vee \neg(a_X \text{is } \alpha)) \wedge F \cup N \models c'_m\}$$

$$r\text{-}deactivated^e(F,N) = \quad \{n' | \exists (e_X, a_X) \in E_X : n' \in VS \wedge c'_r \in \mathcal{E}_\mathcal{F} \wedge$$
$$F \models ((e_X, a_X) \text{ is } c'_r \vee \neg(a_X \text{is } \alpha))\}$$

$$r\text{-}deactivated^s(F,N) = \quad \{n' | n' \in VS \wedge c'_r \in \mathcal{S}_\mathcal{F} \wedge$$
$$F \models (c'_r \vee \neg(a_X \text{is } \alpha))\}$$

$$\text{s.t.} \quad n' = \langle (a_X, \alpha), c'_a, c'_m, c'_d, c'_r, c'_t \rangle$$

Institutionalizing organizations $\mathcal{M}$oise/ JaCaMo

Facts in $\mathcal{M}$oise organizations: constituted state-status functions:

- Agent playing roles: `play(A,R,G)`

  There are environmental and constitutive states that count as the agent A playing the role R in the group G

- Groups being responsible for schemes: `responsible(G,S)`

  There are environmental and constitutive states that count as the group G being responsible to execute the scheme S.

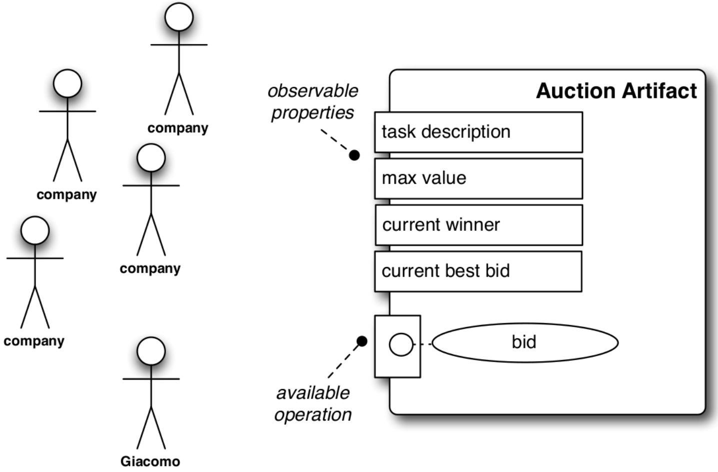
- Agents committed with missions: `committed(A,M,S)`

  There are environmental and constitutive states that count as the agent A being committed to the mission M in the scheme S.

- Achieved goals: `achieved(S,G,A)`

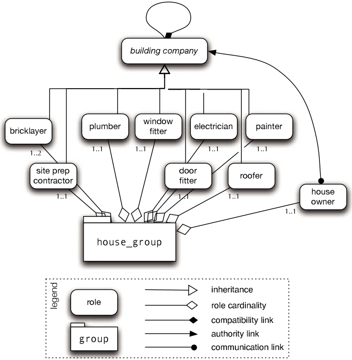
  There are environmental and constitutive states that count as the goal G of the scheme S has been achieved by the agent A.

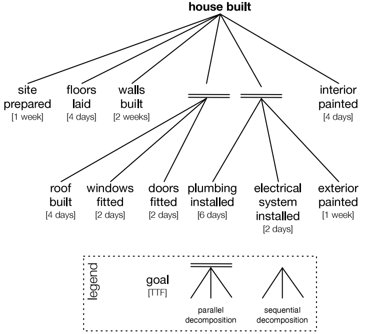
SAI + JaCaMo Example - build a house



Original version available here
SAI version available here

SAI + JaCaMo Example - build a house





norm	role	mission / goals
n1	house owner	house built
n2	site prep contractor	site prepared
n3	bricklayer	floors laid, walls built
...		

SAI + JaCaMo Example - build a house - Role Playing

Original application

1. Giacomo checks the auction winner (by observing an artifact)

   ```
   ?currentWinner(Ag)[artifact_id(ArtId)]
   ```

2. Giacomo contracts the winner

   ```
   .send(Ag, achieve, contract(Task,GroupName))
   ```

3. The winner adopts the corresponding role

   ```
   +!contract(Task,GroupName): task_roles(Task,Role)
         <- adoptRole(Role).
   ```

Giacomo knows the function of the artifact property
depends on another agent to assign institutional function (i.e. the role playing) to the property

SAI Version

```
status_functions:
   states: play(A,R,G)...
constitutive_rules:
   . . .
   2: currentWinner(auction_for_SitePreparation,Agent)
      count-as play(Agent,site_prep_contractor,"hsh_group").
```

The role playing is institucional counterpart of the artifact property

SAI + JaCaMo Example - build a house - Scheme responsibility

Original application

1. Giacomo is responsible to check the group formation
   ```
   ?formationStatus(ok)[artifact_id(GrArtId)];
   ```
2. Giacomo is responsible to assign the scheme to the group
   ```
   addScheme("bhsch")[artifact_id(GrArtId)];
   ```

Giacomo knows organizational details (e.g need for proper formation and scheme assignment)
acts to manage the organization (instead of focus on its goals)
Institutional and organizational roles are mixed (the institution defines what counts as a well formed group)

SAI Version

```
status_functions:
   states: responsible(Group,Scheme),
constitutive_rules:
   . . .
   11: count-as responsible("hsh_group","bhsch")
       while play(Electrician1,electrician,"hsh_group") &
             not(play(Electrician2,electrician,"hsh_group") & not(Electrician1==Electrician2)) &
             play(Site_prep_contractor1,site_prep_contractor,"hsh_group") &
             not(play(Site_prep_contractor2,site_prep_contractor,"hsh_group") & not(Site_prep_contractor1==Site_prep_contractor2))&
             play(Bricklayer1,bricklayer,"hsh_group")&play(Bricklayer2,bricklayer,"hsh_group")&
             not(play(Bricklayer3,bricklayer,"hsh_group")&not(Bricklayer1==Bricklayer3)& not(Bricklayer2==Bricklayer3))&
             play(Plumber1,plumber,"hsh_group") & not(play(Plumber2,plumber,"hsh_group")& not(Plumber1==Plumber2))&
             play(Window_fitter1,window_fitter,"hsh_group") & not(play(Window_fitter2,window_fitter,"hsh_group")& not(Window_fitter1==Window_fitter2))&
             play(Door_fitter1,door_fitter,"hsh_group")& not(play(Door_fitter2,door_fitter,"hsh_group")& not(Door_fitter1==Door_fitter2))&
             play(Roofer1,roofer,"hsh_group") & not(play(Roofer2,roofer,"hsh_group")& not(Roofer1==Roofer2)) &

             play(House_owner1,house_owner,"hsh_group")& not(play(House_owner2,house_owner,"hsh_group")& not(House_owner1==House_owner2)).
```

The group being responsible for a scheme is an institutional consequence of the institutional state

SAI + JaCaMo Example - build a house - Goal Achievement

Original Application

Beyond acting to achieve the goal, the company agents

- must know details on the organizational infrastructure
- manage the organizational infrastructure to inform the goal achievement

```
+obligation(_,_,achieved(bhsch,site_prepared,_),_) : [artifact_id(ArtId)]
  <- prepareSite;
     goalAchieved(site_prepared)[artifact_id(ArtId)].
```

SAI Version

Company agents

- reason about the constitutive rules to achieve the assigned goal
- act in the environment to achieve their assigned goal

```
+obligation(Ag,R,achieved(Scheme,Goal,Ag),_): constitutive_rule(X,achieved(Scheme,Goal,Ag),ToDo,M)
   <- ToDo.
```

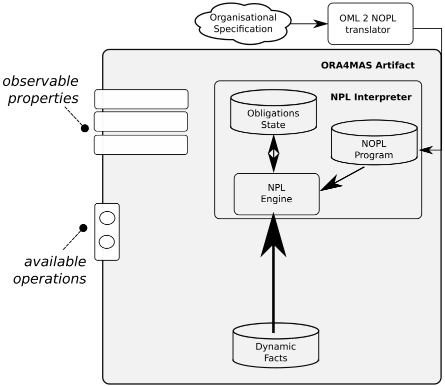
The institution connects the performed action with the goal achievement

```
status_functions:
   states: ..., achieved(S,G,A).
constitutive_rules:
   . . .
   22: count-as achieved(bhsch,site_prepared,Agent)
       when prepareSite[sai__agent(Agent)].
```

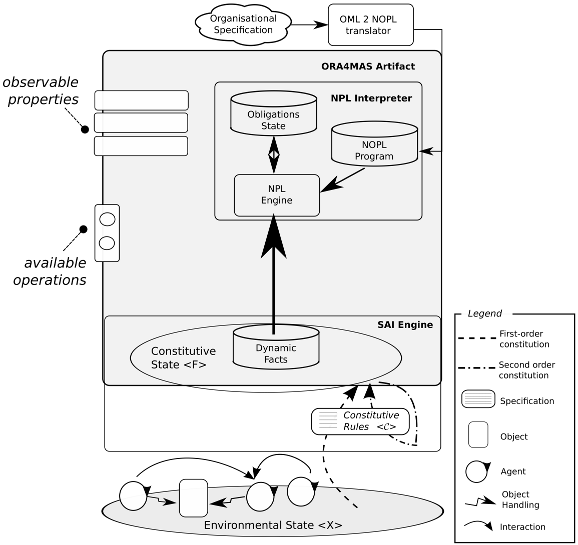
SAI + JaCaMo Example - build a house - Institutional Setup

```
mas house_building {
   agent giacomo
   agent companyA
   agent companyB

   workspace auction{
      artifact auction_for_SitePreparation: AuctionArt("SitePreparation", 2000)
      artifact auction_for_Floors: AuctionArt("Floors",1000)
      artifact auction_for_Walls: AuctionArt("Walls", 1000)
                          ⋮
   }
   institution house_institution: constitutive.sai{
       workspaces: auction
   }
   organisation house_organisation : house-os.xml {
      group house_group: house_group
      scheme house_scheme: build_house_sch
      situated: house_institution
   }

}
```

SAI + JaCaMo ORA4MAS vs. SAI



SAI + JaCaMo ORA4MAS vs. SAI



Future work

- New kinds of status functions (e.g. object-status functions)
- Institutionalization of new social abstractions (e.g. commitments, Accountability)

Final remarks

SAI available at
`github.com/artificial-institutions/sai`

constitutive language interpreter, interfaces for normative platforms, examples, etc.

Publications I

1. BRITO, M. de; HÜBNER, J. F. **Institutional Situatedness in Multi-Agent Systems**. WESAAC 2014. Porto Alegre - Brazil.

2. BRITO, M. de; HÜBNER, J. F.; BOISSIER, O. **A conceptual model for situated artificial institutions**. CLIMA XV. 2014. Prague - Czech Republic.

3. BRITO, M. de; HÜBNER, J. F.; BOISSIER, O. **Bringing constitutive dynamics to situated artificial institutions**. EPIA 2015. Coimbra - Portugal.

4. BRITO, M. de; HÜBNER, J. F.; BOISSIER, O. **Coupling regulative and constitutive dimensions in situated artificial institutions**. EUMAS 2015. Athens - Greece.

Publications II

5. BRITO, M. de; THÉVIN, L.; GARBAY, C.; BOISSIER, O; HÜBNER, J. F. **Situated artificial institution to support advanced regulation in the field of crisis management**. PAAMS 2015. Salamanca - Spain.

6. BRITO, M. de; THÉVIN, L.; GARBAY, C.; BOISSIER, O; HÜBNER, J. F. **Situated regulation on a crisis management collaboration platform**. PAAMS 2015. Salamanca - Spain.

7. BRITO, M. de; THÉVIN, L.; GARBAY, C.; BOISSIER, O; HÜBNER, J. F. **Institution artificielle située pour une aide à la régulation dans le cadre de la gestion de crises**. JFMSMA 2015. Rennes - France.

8. BRITO, M. de; THÉVIN, L.; GARBAY, C.; BOISSIER, O; HÜBNER, J. F. **Supporting flexible regulation of crisis management by means of situated artificial institutions**. Frontiers of Information Technology & Electronic Engineering (FITEE), v.17, n.4.

Publications III

9. BRITO, M. de; THÉVIN, L.; GARBAY, C.; BOISSIER, O; HÜBNER, J. F. **Institution artificielle située pour une aide à la régulation de la gestion de crises**. Revue d'Intelligence Artificielle (RIA), v.30, n.1-2.

10. BRITO, M. de; HÜBNER, J. F; BOISSIER, O. **Situated Artificial Institutions: Stability, consistency, and flexibility in the regulation of agent societies**. JAAMAS.

11. BRITO, M. de; HÜBNER, J. F; BOISSIER, O. **Architecture of an Institutional Platform for Multi-Agent Systems.** PRIMA 2017: 313-329

References I

Huib Aldewereld, Sergio Álvarez Napagao, Frank Dignum, and Javier Vázquez-Salceda.
Making norms concrete.
In Wiebe van der Hoek, Gal A. Kaminka, Yves Lespérance, Michael Luck, and Sandip Sen, editors, *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010): Volume 1-3*, pages 807–814. IFAAMAS, 2010.

Maiquel de Brito, JomiFred Hübner, and Olivier Boissier. A conceptual model for situated artificial institutions. *Computational Logic in Multi-Agent Systems*, volume 8624 of *LNCS*. 2014.

Guido Boella and Leendert van der Torre.
A logical architecture of a normative system.
In Lou Goble and John-Jules Ch. Meyer, editors, *Deontic Logic and Artificial Normative Systems (DEON 2006)*, volume 4048 of *LNCS*, pages 24–35. Springer, 2006.

Guido Boella and Leendert W. N. van der Torre.
Regulative and constitutive norms in normative multiagent systems.
In Didier Dubois, Christopher A. Welty, and Mary-Anne Williams, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004)*, pages 255–266. AAAI Press, 2004.

References II

Cristiano Castelfranchi
Engineering Social Order
In: Omicini A., Tolksdorf R., *Zambonelli F. (eds) Engineering Societies in the Agents World. ESAW 2000*. Lecture Notes in Computer Science, vol 1972. Springer, Berlin, Heidelberg

COMMONS, J. R.
Institutional Economics – Its Place in Political Economy. [S.l.]: The Macmillan Company, 1934. (Institutional Economics Series, v. 1).

CRAWFORD, S. E. S.; OSTROM, E. A Grammar of Institutions.
The American Political Science Review, v. 89, n. 3, p. 582–600, 1995.

Mehdi Dastani, Davide Grossi, John-Jules Ch. Meyer, and Nick A. M. Tinnemeier.
Normative multi-agent programs and their logics.
In Guido Boella, Pablo Noriega, Gabriella Pigozzi, and Harko Verhagen, editors, *Normative Multi-Agent Systems*, volume 09121 of *Dagstuhl Seminar Proceedings*, 2009.

References III

Dastani, M., van der Torre, L., Yorke-Smith, N.
Commitments and interaction norms in organisations
Autonomous Agents and Multi-Agent Systems (2017) 31:2017.

Dignum, Virginia
A model for organizational interaction: based on agents, founded in logic
PhD Thesis, 2004

Dignum, Virginia and Aldewereld, Huib and Dignum, Frank
On the Engineering of Multi Agent Organizations
In Weyns, Danny and Müller, Jörg P, editors, *12th International Workshop on Agent-Oriented Software Engineering (AOSE@AAMAS2011)*

Geiger, R. Stuart
The Lives of Bots
Lovink, Geert and Tkacz, Nathaniel, editors, *Critical Point of View: A Wikipedia Reader*,Institute of Network Cultures,2011.

HODGSON, G. M. What are institutions.
Journal of Economic Issues, n. 1, p. 1–25, 2006.

References IV

Jomi F. Hubner, Jaime S. Sichman, and Olivier Boissier
Developing organised multiagent systems using the MOISE$^+$ model: programming issues at the system and agent levels
Int. J. Agent-Oriented Softw. Eng. 1, 3/4 (December 2007), 370-395.

MILLER, S.
Social institutions.
In: Zalta, E. N. (Ed.). *The Stanford Encyclopedia of Philosophy*. Fall 2012. [S.l.: s.n.], 2012.

NORTH, D. C. Economic performance through time.
American Economic Review, v. 84, p. 359–368, 1994.

SEARLE, J. The Construction of Social Reality. [S.l.]: Free Press, 1995.

SEARLE, J. Making the Social World:The Structure of Human Civilization. [S.l.]: Oxford University Press, 2009.

References V

Shoham, Y., Tennenholtz, M.
On the synthesis of useful social laws for artificial agent societies(preliminary report).
In Proceedings of the 10th national conference on artificial intelligence. San Jose, CA, July 12–16, 1992 (pp. 276–281).

Munindar P. Singh
An Ontology for Commitments in Multiagent Systems
Artificial Intelligence and Law 7:97.

Sofia Panagiotidi, Sergio Álvarez-Napagao, and Javier Vázquez-Salceda.
Towards the norm-aware agent: Bridging the gap between deontic specifications and practical mechanisms for norm monitoring and norm-aware planning.
In *COIN IX*, pages 346–363, 2013.

LUCK, M. et al. Agent technology: computing as interaction (a roadmap for agent based computing). University of Southampton, 2005.

BRITO, M. de; HÜBNER, J. F.; BORDINI, R. H. Programming institutional facts in multi-agent systems. In: ALDEWERELD, H.; SICHMAN, J. S. a. (Ed.). Coordination, Organizations, Institutions, and Norms in Agent Systems VIII. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, (Lecture Notes in Computer Science, v. 7756). p. 158–173.

References VI

DASTANI, M.; TORRE, L. W. N. van der; YORKE-SMITH, N. Monitoring interaction in organisations. In: Coordination, Organizations, Institutions, and Norms in Agent Systems VIII - 14th International Workshop, COIN 2012, Held Co-located with AAMAS 2012, Valencia, Spain, June 5, 2012, Revised Selected Papers. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 17–34.

BOELLA, G.; TORRE, L. van der. Constitutive norms in the design of normative multiagent systems. In: TONI, F.; TORRONI, P. (Ed.). Computational Logic in Multi-Agent Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, (Lecture Notes in Computer Science, v. 3900). p. 303–319. ISBN 978-3-540-33996-0.

CLIFFE, O.; VOS, M. D.; PADGET, J. Answer set programming for representing and reasoning about virtual institutions. In: INOUE, K.; SATOH, K.; TONI, F. (Ed.). Computational Logic in Multi-Agent Systems. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, (Lecture Notes in Computer Science, v. 4371). p. 60–79.

HÜBNER, J. F.; BOISSIER, O.; BORDINI, R. H. A normative programming language for multi-agent organisations. Annals of Mathematics and Artificial Intelligence, v. 62, n. 1-2, 2011.

References VII

- Sébastien Kubicki, Sophie Lepreux, and Christophe Kolski. Rfid-driven situation awareness on tangisense, a table interacting with tangible objects. *Personal and Ubiquitous Computing*, 16(8):1079–1094, 2012

- Lauren Thévin, Fabien Badeig, Julie Dugdale, Olivier Boissier, and Catherine Garbay. Un système multi-agent normatif pour la collaboration et l'interaction mixte. In *JFSMA 18*, pages 1–10, 2014.

- CRAWFORD, S. E. S.; OSTROM, E. A Grammar of Institutions. The American Political Science Review, v. 89, n. 3, p. 582–600, 1995.

- LÓPEZ, F. L. y; LUCK, M.; D'INVERNO, M. A normative framework for agent-based systems. In: BOELLA, G.; TORRE, L. W. N. van der; VERHAGEN, H. (Ed.). Normative Multi-agent Systems. Dagstuhl, Germany: Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2007. (Dagstuhl Seminar Proceedings, v. 07122).

- PIUNTI, M. et al. Embodied organizations: a unifying perspective in programming agents, organizations and environments. In: Proceedings of The Multi-Agent Logics, Languages, and Organisations Federated Workshops (MALLOW 2010). [S.l.]: CEUR-WS.org, 2010. (CEUR Workshop Proceedings, v. 627).

References VIII

- CAMPOS, J. et al. Formalising Situatedness and Adaptation in Electronic Institutions. In: HÜBNER, J. et al. (Ed.). Coordination, Organizations, Institutions and Norms in Agent Systems IV. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, (Lecture Notes in Computer Science, v. 5428). p. 126–139.
- DASTANI, M. et al. Knowledge Representation for Agents and Multi-Agent Systems. In: MEYER, J.-J. C.; BROERSEN, J. (Ed.). Berlin, Heidelberg: Springer-Verlag, 2009. cap. Normative Multi-agent Programs and Their Logics, p. 16–31.
- ALDEWERELD, H. et al. Making norms concrete. In: HOEK, W. van der et al. (Ed.). Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010): Volume 1-3. Richland, SC: IFAAMAS, 2010. p. 807–814.
- FORNARA, N. et al. Artificial institutions: a model of institutional reality for open multiagent systems. Artificial Intelligence and Law, v. 16, n. 1, 2008.
- VIGANÒ, F.; COLOMBETTI, M. Model checking norms and sanctions in institutions. In: SICHMAN, J. S. a. et al. (Ed.). Coordination, Organizations, Institutions, and Norms in Agent Systems III. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, (Lecture Notes in Computer Science, v. 4870). p. 316–329.

References IX

- CARDOSO, H. L.; OLIVEIRA, E. C. Institutional reality and norms: Specifying and monitoring agent organizations. Int. J. Cooperative Inf. Syst., v. 16, n. 1, p. 67–95, 2007.
- BOISSIER, O. et al. Multi-agent oriented programming with jacamo. Science of Computer Programming, Elsevier North-Holland, Inc., Amsterdam, The Netherlands, v. 78, n. 6, p. 747–761, 2013.
- BRITO, M. et al. Situated Artificial Institution to Support Advanced Regulation in the Field of Crisis Management, Advances in Practical Applications of Agents, Multi-Agent Systems, and Sustainability: The PAAMS Collection - 13th International Conference, PAAMS 2015, Salamanca, Spain, June 3-4, 2015.
- Maiquel de Brito, Jomi Fred Hübner, Olivier Boissier: Bringing Constitutive Dynamics to Situated Artificial Institutions. EPIA 2015: 624-637
- BRITO, M. de; HÜBNER, J. F; BOISSIER, O. Situated Artificial Institutions: Stability, consistency, and flexibility in the regulation of agent societies. JAAMAS. 2018
- BRITO, M. de; HÜBNER, J. F; BOISSIER, O. Coupling the normative regulation with the constitutive state management in Situated Artificial Institutions. Knowl. Eng. Rev. 34: e21 (2019)

References X

- BALDONI, M.; BAROGLIO, C.; MICALIZIO, R.; TEDESCHI, S. Accountability as a Software Engineering Tool for Multi-Agent Systems, Legal Knowledge and Information Systems – JURIX 2025: The Thirty-Eighth Annual Conference, Turin, Italy, December 9–11, 2025. Frontiers in Artificial Intelligence and Applications, v. 416, p. 270–275, IOS Press, 2025.
- GARCÍA-CAMINO, A.; RODRÍGUEZ-AGUILAR, J. A.; SIERRA, C.; VASCONCELOS, W. W. Constraint Rule-Based Programming of Norms for Electronic Institutions, Autonomous Agents and Multi-Agent Systems, v. 18, n. 1, p. 186–217, 2009.
- ROCHA COSTA, A. C. Elements for the Agent-Based Modeling of Slavery Systems, ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, v. 9, 2020.
- HÜBNER, J. F.; BOISSIER, O.; KITIO, R.; RICCI, A. Instrumenting Multi-Agent Organisations with Organisational Artifacts and Agents, Autonomous Agents and Multi-Agent Systems, v. 20, n. 3, p. 369–400, 2009. DOI: 10.1007/s10458-009-9084-y.